Automatic Sentiment Analysis Using the Textual Pattern Content Similarity in Natural Language

Jan Žižka and František Dařena

Department of Informatics – SoNet Research Center
Faculty of Business and Economics, Mendel University in Brno
Zemědělská 1, 613 00 Brno, Czech Republic
{zizka,darena}@mendelu.cz

Abstract. The paper investigates a problem connected with automatic analysis of sentiment (opinion) in textual natural-language documents. The initial situation works on the assumption that a user has many documents centered around a certain topic with different opinions of it. The user wants to pick out only relevant documents that represent a certain sentiment – for example, only positive reviews of a certain subject. Having not too many typical patterns of the desired document type, the user needs a tool that can collect documents which are similar to the patterns. The suggested procedure is based on computing the similarity degree between patterns and unlabeled documents, which are then ranked according to their similarity to the patterns. The similarity is calculated as a distance between patterns and unlabeled items. The results are shown for publicly accessible downloaded real-world data in two languages, English and Czech.

Keywords: sentiment/opinion analysis, textual patterns, natural language, textual document similarity, similarity ranking.

1 Introduction

In this article, the authors present a procedure how to simply single out textual documents representing a specific opinion from a big group of documents with different opinions. The whole document group is centered around one common topic. This problem in question is now very actual. The last years have introduced various new web-technologies as Web 2.0, which facilitates a massive expansion of using the Internet for expressing different or miscellaneous opinions (via messages) between people, groups of people, administrative authorities, commercial or non-commercial organizations and institutions, and so like. Shared on-line journals known as *blogs* enable people to post daily entries about their personal experiences and hobbies. Customers can provide valuable feedbacks both for e-shops (like *amazon.com*) and their potential future customers who can read opinions and experiences willingly provided by shoppers. Similarly, the Internet is a host of uncountable discussion groups, newsgroups, and tens of social networks like *Facebook* that unites more than 100 millions of unique visitors. The significant part of the reciprocal communication data is expressed in natural languages using the textual form. Such the data are very interesting because they usually hide a lot of information and knowledge that can be used for various goals,

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 224–231, 2010.
© Springer-Verlag Berlin Heidelberg 2010

including the business ones. Therefore, the question is how to mine that textual data. During the last 20 years or so, many useful text-mining methods and algorithms have been developed [1]. If the data enable the application of classification, it is the best approach. However, very often the potential training data miss labeling that is necessary for the future including of individual items (text documents) into their appropriate classes [2].

2 Basic Ideas

The article points to an alternative method that avoids the large labeling of individual training items. This method, described in the next chapters, introduces more initial information into the process of recognizing unlabeled items. The idea comes from a typical real situation when somebody has only a small collection of 'good' examples, or *patterns*, available. From a large number of various text documents, he or she wants to remove what is different from the patterns so that the remaining part of textual items represents interesting or relevant group of articles, blog entries, or discussion submissions. The main idea is based on determining the *similarity* degree [3] between the patterns and an unlabeled textual item [4]. In addition, this approach is not aimed at categorizing the items according some given topics. Typically, there is only one main shared topic and the authors are interested in separating the unlabeled text items according different *opinions* (or, as it is often used as an alternative term, *sentiments*). The automatic sentiment/opinion analysis can sort those numerous individual contributions that are expressed in natural languages. For example, people purchasing a specific product using an on-line web-based shop can later write their opinions [5]. Other purchasers can then submit similar or different meanings, also with a reference to previous entries, and the e-seller can analyze such submissions and draw appropriate conclusions. Similarly, in various discussion groups, some people can provide a message about their opinion of a book, film, politician, and so like – it can help others for their final decisions if yes or no. Thus, a reader of such text messages may want to select only positive or negative items from a large collection. He or she can choose some patterns which correspond to his or her sentiment and then let a machine find similar items.

3 Data Sets Used for Experiments

The main idea of looking for documents that are similar to patterns was tested using several real-world data sets obtained by downloading from publicly and commonly accessible data sources: *amazon.com* [6] plus *amazon.co.uk* [7] (in English), and the archive of a Czech newspaper *MF Dnes* articles (the electronic version) [8] plus a web-site of a Czech political party [9] (both in Czech). The selected *amazon* data included opinions of customers on a particular iPod head-phones (consumer electronics) – 30 positive and 48 negative customer-opinions; computer hardware (Western Digital USB external hard disk) – 47 positive and 37 negative opinions; a book (King James version of Bible) – 130 positive and 86 negative opinions; a film (Caligula, the Unrated Edition)

– 100 positive and 64 negative opinions; and articles on a political party program (the Czech Social Democratic Party, ČSSD) – 22 positive and 22 negative opinions.

The *amazon* data-sets consist of various entries describing consumers' meanings. In addition, each contributor assigns one star (the worst evaluation of a product) up to five stars (the best evaluation) according to his or her opinion. The authors selected only two groups: with one- (negative) and five-star (positive) evaluation, excluding the rest because 2, 3, and 4 stars expressed more-or-less ambiguous opinions – it was also difficult for humans to decide whether it was a positive or negative opinion as those opinions were more-or-less mixed. The goal was an attempt to automatically select contributions belonging to one- or five-star evaluation using only the text contents.

It is necessary to remark that the *amazon* data are typical, with many expected problems. First, some contributions are long enough to clearly express the opinion, but most of them are short, some of them very short. Second, some positive (or negative) contributions express the same opinion using different words, for example, synonyms, idioms, or word connections – it means that such documents can look different from the word-based viewpoint. Third, the contributions contain mistypings and spelling errors, therefore the same word typed differently by two persons looks absolutely different from a machine point of view, extending the dictionary. To name but a few from, for example, the Western Digital USB hard-disk data: *absoloutely/absolutley/absolutely*, *beleive/believe*, *definately/definitely,* where only the last word form is correct. Anyway, these problems are quite typical (and naturally inevitable) for such blogs and discussion groups. The decision to experiment also with Czech texts was intended to show whether the presented method works also for other language than English. As an example, the authors found politically-based data that contained contributions *for* and *against* one of the present-day Czech political party platforms before elections. The realization and results of experiments with all the above mentioned data sets are described in the following sections.

4 Text Document Pre-processing and Representation

Textual documents in a natural language contain words and these words are used for determining the similarity between two documents. The preceding research of many scientists brought a lot of algorithms and methods that proved their good quality and usefulness also in practice. A standard procedure includes pre-processing based on creating a *bag-of-words* and consecutively a *dictionary* (a set of words) from words in text documents available. A certain advantage is that such a process can be easily done by computers. On the other hand, a bag-of-words contains words without the information about their original positions in a document – a loss of information. More sophisticated methods need more complex procedures that are – unlike the simple bag-of-words approach – much more dependent on a specific language. The authors decided to employ the standard procedure because they wanted to create a method based on the automatic natural-language processing without a big dependency on individual languages.

The pre-processing did not use removing stop-words because most of the items were very short and it would be necessary to carry out the specific stop-word analysis for

each of the data collection – not always the list of common stop-words can be used. The shortness of text items was also the reason why the pre-processing procedure did not remove words with too high or too low frequency. The authors do not exclude the possibility that a deeper analysis of the text properties could somehow improve the results but it was not the primary goal of this paper.

Analogously, the 'short' words (typically, up to three letters) were not removed as it is typical for English texts because the authors wanted to apply the same approach to another (Slavonic) language, as well. Such a procedure would request additional text analysis that could be specific for different languages.

A word can be represented by several popular methods [1]: as a binary number (1/0, or a word is/is not in a document), frequency number (how many times a word is in a document), or $TF \times IDF$ (term frequency times the inverted document frequency). Independently on the representation, the dictionary (as well as each document) was transformed into a multidimensional vector where individual word representations were used as coordinates within the abstract space with each dictionary word as one of dimensions. Several initial experiments showed that the best results were provided by the frequency representation.

It is necessary to emphasize that the vectors did not contain the original class membership (that is, belonging to the positive or negative group of opinions).

5 The Similarity of Unlabeled Documents to Patterns

The similarity degree between an unlabeled document and a selected pattern is computed using the word representation. Here, the word frequency representation was used as features in the vectors, where a vector represents one text document. The more words with their frequencies are equal in a pair of vectors the higher the similarity between both documents is, and vice versa. Each document can be taken as a point in an n-dimensional abstract space with coordinates given by the word frequencies (each word makes one of the axes). In this case, the similarity degree can be expressed as a distance between two points where the zero distance means an identical couple of documents [1]. Therefore, having two textual documents A and B, the Euclidean distance L_E between a text document $d^{(A)}$ and $d^{(B)}$ in Cartesian coordinates can be calculated in the following way:

$$L_E = \sqrt{\sum_{i=1}^{n} (d_i^{(A)} - d_i^{(B)})^2}\,, \tag{1}$$

where $d_i^{(.)}$ is the i-th word coordinate of an n-dimensional point representing a document, and n is the number of unique words in the dictionary of all documents used as examples. The values of $d_i^{(.)}$ are the mentioned word frequencies.

If a document is taken as a vector in Cartesian coordinate system, then the vector similarity is given by the angle α between both vectors, where the zero angle means the 100% similarity. Typically, the vectors are very sparse, containing only few shared words between documents A and B. The similarity is expressed using *cosine* of the angle α:

$$cos(\alpha) = \frac{A \cdot B}{\| A \| \| B \|}, \tag{2}$$

which is a dot-product of two vectors A and B using the word frequencies as the features, where $\| \, . \, \|$ is a vector magnitude (length).

The other question is how many patterns are necessary as the good samples of an opinion which should be extracted from the whole data collection. It depends on the available number of examples that express the monitored kind of opinion. Too low example number may restrict the extracted documents too much. Too many different examples of a certain opinion may be the leading cause of too broad extraction with too many documents having a low similarity. Or, if there are too many very similar patterns, it can lead to a high and redundant computational complexity coming out from computation of many distances.

6 Ranking by Similarity

After computing distances of all unlabeled text documents to the patterns, the documents are sorted so that the most similar one is at the top of the rank and the less similar ones are placed lower. Then, the user may decide how many documents from the rank top could be processed by any following method. Typically, a user chooses a relatively small number, units or tens, of the top-ranked items for his or her following work. The rank can contain thousands or far more documents but the user can process only a limited number of them – naturally, the user wants the items that are very similar to the known patterns. Such a procedure is similar to 'classification' when only one class is known while all the available unlabeled items can belong to the unknown number of various classes.

As it can be expected, the suggested procedure cannot be errorless. The error of the similarity ranking originates in the fact that at the top of the rank there can also be false positives because the similarity is measured imperfectly and the noisy text items may sometimes cause improper results. Testing the procedure with known data, it is possible to study how many correct items are at the top and how the correctness decreases towards the bottom. Ideally, all items with the monitored opinion should be higher than items with different opinions. However, as the similarity decreases, the possibility of errors increases – in fact, also human beings often cannot quite unambiguously decide if an item still belongs among the relevant ones or not.

7 Experiments and Their Results

In the experiments with the five data sets mentioned above, randomly selected subsets of textual items with a specific opinion were used as patterns. The number of patterns changed from 10 to 35 (for details, see also the graph in Figure 1 and Table 1 description), depending also on the size of the original data set to keep the sufficient number of testing examples. For the *amazon* data, the authors focused on the radical univocal opinion degrees: to reveal whether it would be simply possible to correctly assign positive opinions written in a natural language to the five-star (or one-star)

category. After detaching the patterns, the rest of the opinion examples was used for testing – the set of unlabeled items containing both the same monitored opinion, and the unmonitored rest (the opposite opinion).

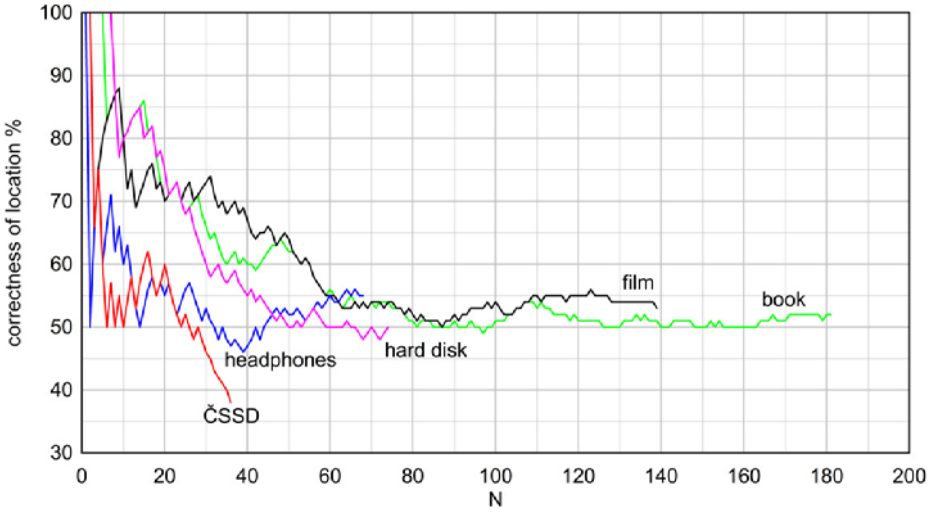

Fig. 1. With the increasing number N of documents taken from the top of the rank, the chance of obtaining misplaced items increases as well. At the beginning, for $N < 10$, some curves overlap but all start at $N = 1$. The ČSSD curve is for the political discussion data in Czech. The remaining four curves demonstrate positive and negative opinions in English written by *amazon* customers on various purchased products.

Applying the similarity procedure described above, the degree of errors was studied in dependence on the number of patterns and the type of the similarity as the parameters. In the graph Fig. 1, a reader can see the results for the optimal parameters. The experiments used the two similarity methods (*Euclidean* and *cosine*) with a small negligible difference between them. In some cases the *Euclidean* distance provided slightly better results than the *cosine* one. Thus, only the results with the *Euclidean* similarity are demonstrated here.

Each member of the testing set was successively compared to all the individual patterns. As the final similarity degree, the nearest pattern was taken, where the degree was given by the computed distance. After processing of all the testing samples, they were ranked according their similarity degree starting with the most similar ones at the top. Then, the number of misplaced items specified the error. Ideally, all the samples with the monitored opinion should be placed before the first item having the opposite opinion. However, some items were misplaced, especially those ones that were too brief, having not many words. The experiments showed that larger texts provided lower errors, which could be logically expected because of the higher information contents given by words. The curves in the graph Fig. 1 depict the percentage of correct opinion-placing (the vertical coordinate axis *correctness of location*) for the first N documents in

Table 1. Parameters of the individual data groups. The table shows the total number of samples, the number of samples with the positive (+) and negative (−) opinion, and the number of patterns with either the positive or negative monitored opinion. The negative monitored opinion took patterns from the negative samples and vice versa.

data group	*samples*	*+ opinions*	*− opinions*	*patterns*	*monitored opinion*
headphones	78	30	48	10	−
hard disk	84	47	37	10	+
book	216	130	86	35	+
film	174	110	64	25	+
ČSSD	44	22	22	8	+

the rank (N, the horizontal axis). For example, if a user would select the first five items ($N = 5$) from the top, there may be only examples of his or her monitored opinion type: the correctness would be 100 %. Analogously, for the first $N = 10$ items with three misplaced ones, the correctness is 70 %. As the N increases, the correctness expectedly decreases because the similarity of all samples varies and sometimes overlaps for different opinions.

The table Tab. 1 shows the detailed parameters for each of the five individual data groups. It is obvious that the data sets having more examples for both of the investigated opinion groups (positive and negative) provided better results: see the curves *book* and *film*. Such the data could give also more patterns for the monitored opinion (35 and 25, resp.). Clearly, their correctness descent is slower than for the other, smaller data. The political data gave the best results for 8 patterns, which is obviously not too many comparing with the other data groups, and it would be necessary to collect more examples for both opinions. The *headphones* and *hard disk* data are somewhere in the middle, which corresponds to the number of available examples.

8 Conclusions

The experiments with the five groups of textual data in natural language downloaded from the real world demonstrated that the procedure based on selecting a limited number of good patterns representing a certain opinion centered around a specific topic can provide acceptable results. The unlabeled samples are ranked according to their similarity with the patterns. It is up to a user how many of the most similar samples he or she takes for the future usage – with the increasing number of selected items the error of an inappropriate selection increases. The results also confirmed that more patterns increase the correctness of the selection. It is necessary to emphasize the fact that the goal was not to exactly separate classes – it would be a task for a classification procedure providing that all the training samples are labeled. A request from potential users is to select not too many (max. tens) 'good' items from hundreds, thousands, or much more. In such a case, manual labeling of all training examples would be too much expensive, if realizable. The continuing research is focused on more elaborate data pre-processing to improve the selection correctness for more and larger data collections. In addition, the study of dependence on different similarity algorithms is planned as well.

Acknowledgements. The research work published in this paper was supported by the Research program of Czech Ministry of Education VZ MSM 6215648904.

References

1. Srivastava, A.N., Sahami, M.: Text Miming: Classification, Clustering, and Applications. Chapmann and Hall/CRC, New York (2009)
2. Hroza, J., Žižka, J.: Mining Relevant Text Documents Using Ranking-Based k-NN Algorithms Trained by Only Positive Examples. In: Proceedings of Knowledge 2005, pp. 29–40. VŠB-Technical University, Ostrava (2005)
3. Hroza, J., Žižka, J.: Selecting Interesting Articles Using Their Similarity Based Only on Positive Examples. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 608–611. Springer, Heidelberg (2005)
4. Žižka, J., Hroza, J., Pouliquen, B., Ignat, C., Steinberger, R.: The Selection of Electronic Text Documents Supported by Only Positive Examples. In: Proceedings of the 8th International Conference on the Statistical Analysis of Textual Data, JADT 2006, Besançon, France, April 19-21, pp. 993–1002. Presses Universitaires de Franche-Comte (2006)
5. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 2004, Seattle, Washington, August 22-25. ACM, New York (2004)
6. Amazon USA (March 2010), `http://www.amazon.com`
7. Amazon UK (March 2010), `http://www.amazon.co.uk`
8. MF Dnes (March 2010), `http://mfdnes.newtonit.cz`
9. ČSSD (March 2010), `http://www.cssd.cz`